CLAIMS

What is claimed is:

1. A cache memory comprising:

a cache buffer;

a storage array comprising a plurality of cache memory locations and selectively receiving data from said cache buffer, selectively received said data being stored in ones of said memory locations; and

a tag memory storing tags associated with data in said storage array and selected data in said cache buffer.

2. A cache memory as in claim 1, wherein said cache input data selectively includes executable commands.

3. A cache memory as in claim 1, wherein said cache buffer comprises:

a cache input buffer receiving cache input data.

4. A cache memory as in claim 3, wherein said cache buffer further comprises:

an output buffer containing most recently accessed data, ones of said tags in said tag memory associated with said most recently accessed data.

5. A cache memory as in claim 4, wherein said tag memory comprises:

a first content addressable memory (CAM) containing tags associated with data stored in said storage array; and

a second CAM containing tags associated with said most recently accessed data.

6. A cache memory as in claim 5, wherein a tag for requested data is checked against tags in said second CAM and said cache input buffer before checking tags in said first CAM.

7. A cache memory as in claim 5, wherein each of said first CAM and said second CAM are a circulating first in first out register (FIFO).

8. A cache memory as in claim 4, wherein each said storage array is a static random access (SRAM) array.

9. A cache memory as in claim 1, wherein cache power is substantially less for accessing said data in said cache buffer than for accessing data in said storage array.

10. A content addressable memory (CAM) random access memory (RAM) cache comprising a plurality of CAMRAM banks, each of said CAMRAM banks comprising:

a cache buffer receiving cache input data, said cache input data selectively including executable commands;

a bank store comprising a plurality of cache memory locations and selectively receiving data from said cache buffer, selectively received said data being stored in ones of said memory locations; and

a CAM storing tags associated with data in said storage array and selected data in said cache buffer.

11. A CAMRAM as in claim 10, wherein said cache buffer comprises:

an input buffer line receiving a cache input data line; and

an output buffer containing most recently accessed data, ones of said tags in said CAM being associated with said most recently accessed data.

12. A CAMRAM as in claim 11, further comprising a cache storage buffer, each said input buffer line in said plurality of CAMRAM banks being a line in said cache storage buffer.

13. A CAMRAM as in claim 11, wherein said CAM comprises:

an *n*-CAM having *n* tag locations, each *n*-CAM tag location being associated with one of *n* storage locations in said bank store; and

an *i*-CAM containing *i* tag locations, wherein $n > i$ and each *i*-CAM tag location is associated with a location in said output buffer.

14. A CAMRAM as in claim 13, further comprising means for checking a tag for requested data against tags in said *i* CAM and said cache input buffer independent of tags in said *n* CAM.

15. A CAMRAM as in claim 14, wherein said checking means only checks for said tag in said *n* CAM, when said tag is not found in said *i* CAM or in said cache input buffer.

16. A CAMRAM as in claim 15, wherein cache power is substantially less for accessing said data in said cache buffer than for accessing data in said bank store.

17. A CAMRAM as in claim 13, wherein each of said *n*-CAM and said *i*-CAM are a circulating first in first out register (FIFO).

18. A CAMRAM as in claim 11, wherein said bank store is a static random access (SRAM) array.

19. A method of managing data in a cache, said method comprising the steps of:

a) providing incoming data to an input buffer;

b) selectively loading data from said input buffer into a storage array;

c) selectively loading accessed data from said storage array to an output buffer, a number of most recently accessed data blocks being held in said output buffer; and

d) selectively providing data from each of said input buffer, said storage array and said output buffer responsive to an access request.

20. A method of managing data as in claim 19, said method further comprising the steps of:

e) receiving an access request for data; and

f) checking said input data buffer for data requested for access.

21. A method of managing data as in claim 20, wherein said access request is a store request and said method further comprises the steps of:

g) storing said data in said input buffer; and

h) marking said stored data as dirty.

22. A method of managing data as in claim 20, said method further comprising the steps of:

g) checking said output buffer for said data requested for access.

23. A method of managing data as in claim 22, wherein said access request is a store request and said method further comprises the steps of:

h) storing said data in said output buffer; and

i) marking said stored data as dirty.

24. A method of managing data as in claim 22, wherein said output buffer is checked in step (g) coincident with checking said input buffer in step (f).

26. A method of managing data as in claim 22, wherein whenever said data requested for access is not found in said output buffer or said input buffer, said method further comprises the steps of:

h) checking said storage array for said data requested for access.

27. A method of managing data as in claim 26, wherein whenever said data requested for access is found in said storage array, said method further comprises the steps of:

i) loading said data requested for access into said output buffer; and

j) providing said data requested for access as an output.

28. A method of managing data as in claim 26, wherein whenever said data requested for access is not found in said storage array, said method further comprises the steps of:

i) sending a miss request;

j) loading said input buffer; and

k) providing said data from said input buffer as an output.

29. A method of managing data as in claim 28, wherein whenever said input buffer contains data other than said data requested for access, said sending step (h) further comprises loading other said data from input buffer to said output buffer.

30. A method of managing data as in claim 26, wherein data in each of said input buffer, said storage array and said output buffer are identified by tags, said tags being checked in checking steps (f), (g) and (h).